POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Homework report

Homework 3

Scientific computing tools for advanced mathematical modelling

Authors: Caterina Leimer Saglio, Davide Carrara and Francesco Romeo

Academic year: 2021-2022

---

1. Mathematical formulation of the problem

1.1. Introduction

The aim of the project is to implement a model which is able to predict data related to the COVID19 pandemic in three italian regions: Lombardia, Lazio, Sicilia. For each of these three regions we have a daily dataset containing information about four categories that we aim at predicting: New Infections, Hospitalised, Recovered, Deceased. In particular, we decided to discard data related to 2020, given the fact that the vaccination campaign started at the end of that year.
To accomplish our goal, we implemented two type of models: a neural network and an ARIMA model; at the final stage we used for each category the model that performs the best.

1.2. ARIMA Models

A popular way to work with time series data is by using an ARIMA model, which is a generalization of an ARMA model. In particular, given a time series X_t, an ARMA(p,q) model is described by:

$$X_t - \alpha_1 X_{t-1} - ... - \alpha_p X_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}$$

where ϵ_{t-k} represents a white noise, and the parameters p and q characterise the order of the model.
ARMA models can be deployed to better understand the evolution of time series and to forecast future values, but they can be applied only to stationary time series, i.e. stochastic processes whose mean and covariance don't depend on time. This implies that the time series should not have any pattern or any particular behavior, which is not the case of our data set, since COVID19 pandemic was characterized by different seasonal trends. To recover stationarity of a time series usually it is sufficient to take difference of consecutive observations, and this is what ARIMA models do. Considering a time series X_t, an ARIMA(p,1,q) model is characterised by the following equations:

$$Y_t = X_t - X_{t-1}$$

$$Y_t - \alpha_1 Y_{t-1} - ... - \alpha_p Y_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}$$

The variable Y_t can be seen as an estimate of the first derivative of the time series X_t.
Generalizing to higher order differentiation we obtain ARIMA(p,k,q) model, which can be better specified using the lag operator L, defined as:

$$L^k X_t = X_{t-k}$$

Using this operator the equations of the ARIMA(p,k,q) model can be rewritten in the following way:

$$Y_t = (1 - L)^k X_t$$

$$\left(1-\sum_{i=1}^{p}\alpha_i L^i\right)Y_t=\left(1+\sum_{j=1}^{q}\theta_j L^j\right)\epsilon_t$$

This is the theoretical framework in which we will operate.
The next step is to understand how to fit the ARIMA model to the data. It is clear from the equations above that to do so we should estimate the value of the parameters (p,q,k) and the values of the coefficients (α_i,ϵ_j): the former can be estimated using the Bayesian Information Criterion (BIC), while the latter are the maximum likelihood estimators, which in this setting can be proven to coincide with the least squares estimators. These two steps are intertwined: first it is necessary to fix the values of (p,q,k) in order to know how many parameters should be estimated, then once the estimation is obtained, the BIC can be computed.
The coefficients are thus obtained by solving the following optimization problem:

$$\min_{\alpha,\theta}\sum_{t=1}^{T}[Y_t-\hat{Y}_t(\alpha,\theta)]^2$$

where T represents the maximum time horizon of the problem, i.e. the numerosity of the given data set.
Now we can compute the BIC, which is an index that takes into account the complexity of the model together with its goodness of fit in correspondence of the estimated coefficients. The formula of the BIC is:

$$BIC=-2log(\hat{L})+log(T)(p+k+q)$$

where (indicating with $\vec{x}$ the data observed)

$$\hat{L}=p(\vec{x}|\hat{\alpha},\hat{\theta})$$

is the likelihood function evaluated in the parameters that maximize it, which, as stated above, coincide with the least squares estimators.
The BIC index can be now used to perform model selection: we consider different values for the parameters (p,k,q), fit the corresponding model and compute the related BIC; then, the model associated to the minimum value of the BIC index is selected as our model to fit the data.
This procedure guarantees to obtain the best trade-off between complexity of the model and goodness of fit.

1.3. Neural Network

1.3.1 Choosing Neural Networks

The choice of Neural Network for the problem of predicted COVID-19 related quantities is debatable, since the amount of data at our disposal is limited. On the other hand, Neural Networks are theoretically able to learn any function, and are most of all particularly good in discovering hidden relationships among different quantities. Even if the evolution of COVID has been sometimes difficult to foresee, due to the impact of new variants, it is also true that the cyclic coming and going of *waves* can be learned from a model.

1.3.2 Structure

The Neural Networks we chose to use for the three regions are relatively small in size, in the order of the tens of thousands parameters, with their structure based on sequences of Dense and Batch Normalization layers. We also applied some cautions to prevent overfitting, given the small amount of data, namely dropout and weights regularization.

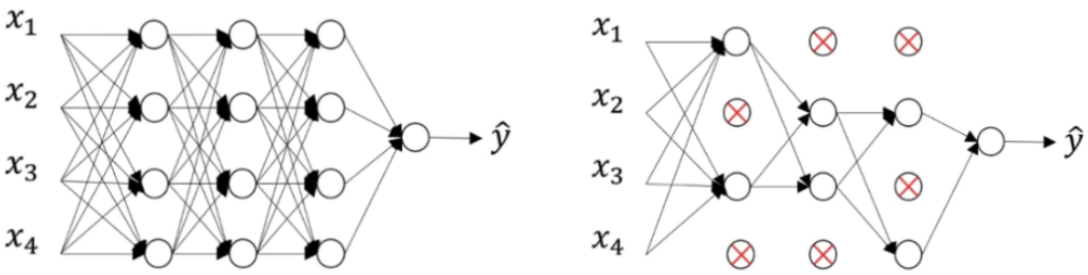


Figure 1: Dropout

Figure 1 gives a representation of the dropout technique. During each training epoch each neuron is randomly switched off with a given probability p, while all of them are used at prediction time.

Weight regularization is a typical technique to prevent overfitting, and consists in adding a term to the loss function which is directly related to the norm of the parameter. In our specific case we applied both L1 and L2 regularization, similarly to ElasticNet Regularization. The new loss is reported here:

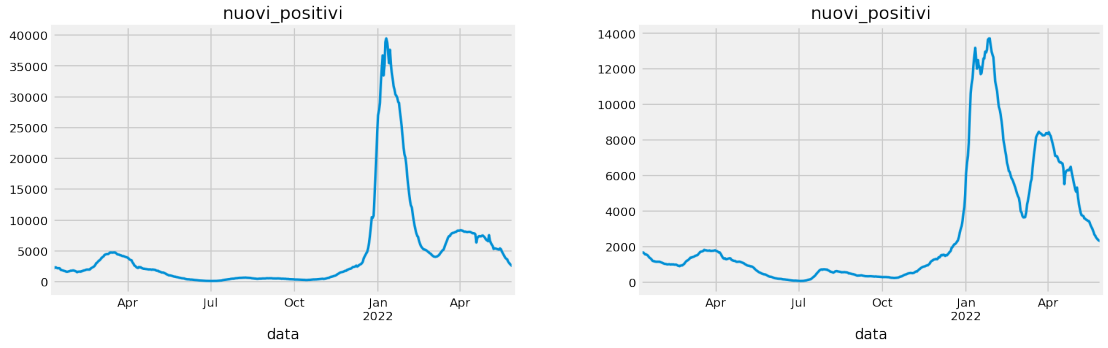
$$\hat{\mathcal{L}}(W) = \frac{\alpha}{2}||W||_2^2 + \beta||W||_1 + \mathcal{L}(W) = \frac{\alpha}{2}\sum_i\sum_j w_{i,j}^2 + \beta\sum_i\sum_j |w_{i,j}| + \mathcal{L}(W)$$

2. Methods

2.1. ARIMA

2.1.1 Pre-processing

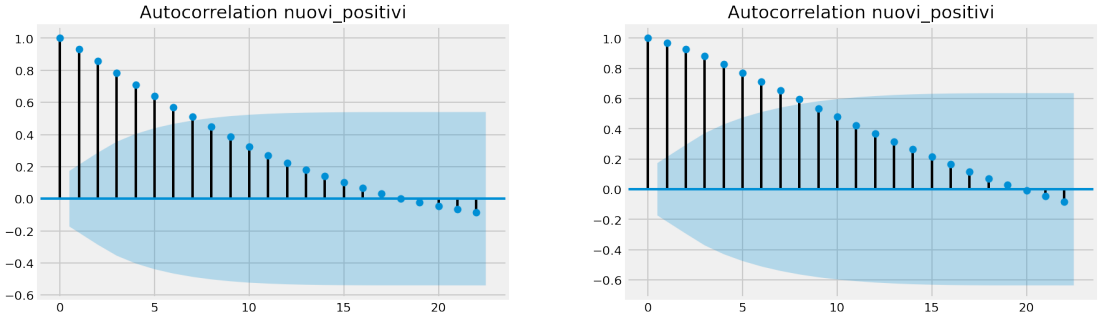
For the analysis of the new positives, since the data are particularly noisy, it was decided to exploit a regularisation by means of a rolling-window, which makes it possible to take into account the 7 days preceding the analysis and to compute an average of these, thus obtaining a much more regular dataset. Figure 10 shows the new dataset found by pre-processing the Lombardia and Lazio regions dataset, where the first day taken into account is '2021-01-02'.



(a) Dataset - Lombardia (b) Dataset - Lazio

Figure 2: Dataset

It is worth to mention that we used the rolling window procedure always on daily data and not on cumulative ones.
To have a first qualitative information about the correlation between the data of one day and the one of the previous days, we plotted the autocorrelation function.
We report only the autocorrelation function for the new infections in Lombardia and Lazio.



(a) Auto-correlation - Lombardia (b) Auto-correlation - Lazio

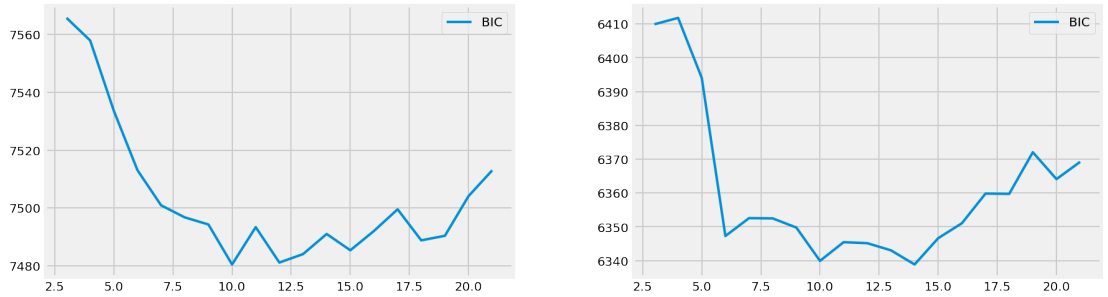
Figure 3: Auto-correlation new positive

The blue band represents the confidence intervals for the estimated autocorrelation. Even if these plots are

influenced by the rolling procedure applied before, they can anyway give a first insight on the number of parameters needed in the ARIMA models we will build.

2.1.2 Optimal choise for the parameters

To obtain the ARIMA model, it is necessary to derive the optimal values of the three coefficients (p, d, q) where p is the order (number of time lags) of the auto-regressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. To perform this estimate, we use the procedure described in Section 1.2, which aims at selecting the model that minimizes the Bayesian Information Criterion (BIC), given that the lower the value of this criteria for a range of models being investigated, the better the model will suit the data.
In particular to draw the plot of the BIC versus the degree of freedom of the model we select, for each degree of freedom, the model with the lowest BIC. In this way we can derive from this plot the model with minimum BIC.
We can see the BIC functions related to the new positives of Lombardia and Lazio in the following plots:



(a) Dataset - Lombardia - new positives (b) BIC - Lazio - new positives

Figure 4: BIC

The models with the minimum BIC have 10 degrees of freedom for Lombardia and 14 for Lazio. In particular, the corresponding values of the parameters of the ARIMA models are displayed in this table:

BIC Values					
	BIC	AR	dif	MA	dof
Lombardia	6338.868067	7.0	1.0	6.0	14.0
Lazio	6426.273195	3.0	1.0	6.0	10.0

Table 1: BIC values

We followed this procedure for all the four categories in the four regions.

2.2. Neural Network

2.2.1 Decomposing and Smoothing

When trying to predict deceased and hospitalized we chose to consider the daily increments, rather than the cumulative quantities. This passage, implicitly computed by the ARIMA model, was our first preprocessing step.
However, considering daily increments leads to extremely noisy data series, due in part to the random nature of the pandemic phenomenon but mostly to bureaucratic aspects in the registration. Specifically, a clear weekly pattern is observable, with more registration during the days of the week and negative peaks along the weekend. In order to reduce the noise effect and ease the learning process of the Neural Network, we decided to apply a seasonal decomposition.

This kind of approach, represented in Figure 5 for the series of new deceased, relies on the assumption that the time series can be expressed by the sum of three main terms: the Trend, the Seasonal term and the Residual term. The trend represents the main dynamic of the series over time, while the Seasonal term is a repetitive pattern over a certain timespan, usually linked to a specific measurement error or an external phenomenon. Lastly, the residual represent the remaining noise, which cannot be predict under these assumption and should ideally be identically distributed over time.

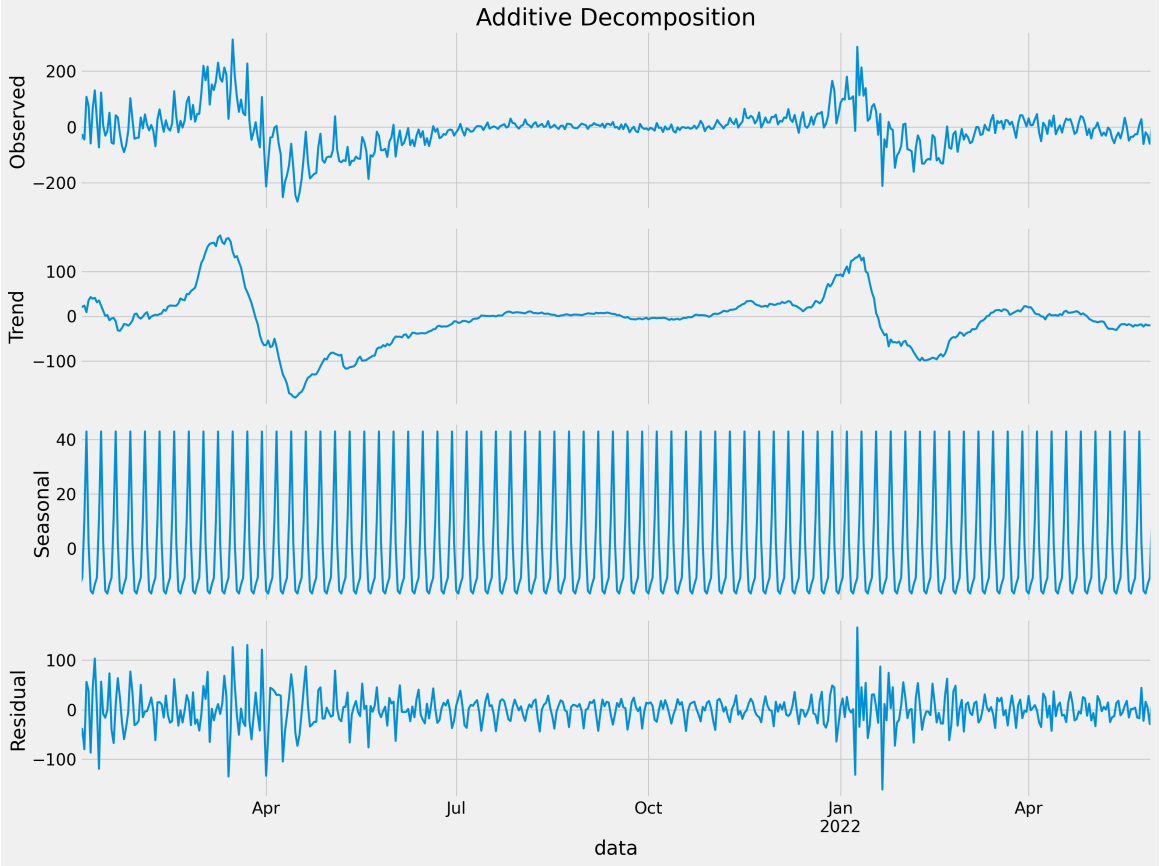


Figure 5: Seasonal Decomposition for daily deceased in Lombardia

As we can see from Figure 5, this approach is particularly suited for the data at our disposal. The identified trend is coherent with the evolution of the pandemic, and the seasonal term has a weekly period, with high peaks during the week and lows during the weekends. The residuals seem to satisfy our distributional assumption, with the notable exception of the Delta and Omicron variant peaks: the latter will be specifically taken care of in the following steps.
In order to additionally increase smoothing of the data, a rolling window smoother with window size equal to 5 is used on the data series.

2.2.2 Differentiation

The simple collection of daily deceased and hospitalized is not sufficient to fully represent the phenomenon and proved to be a poor input for our Neural Network. That is why we also provided as input the estimates of first, second and third derivative of these quantities.

2.2.3 Dataset Subdivision

In order to have a meaningful training for our Neural Network, we need to define a training and validation set. The best case scenario would be to consider only the most recent data, since the Omicron variant introduced significant modifications in the magnitude of the reported cases. However, the numerosity of the post-Omicron days is not sufficient for the training: as a consequence we consider data from 01/02/2021 to 05/12/2021 as a

pre-peak training set. The training set is completed from the post-peak part, including days from 05/01/2022 to 06/04/2022. The following days create the validation set.

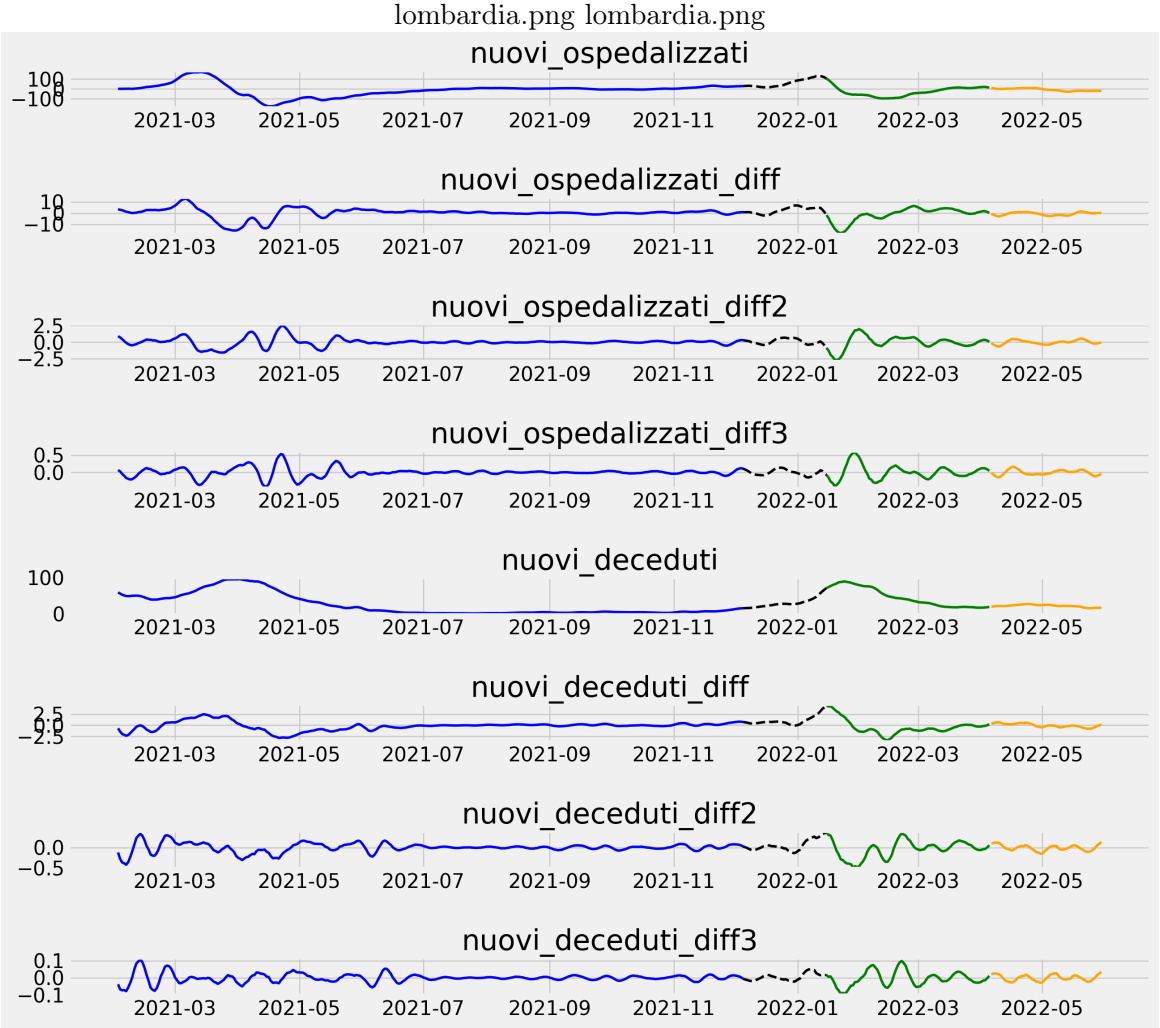


Figure 6: Example of input for the Neural Network for Lombardia

One example of the input dataset for the Neural Network is represented in Figure 6: in blue the pre-peak training, dashed is the peak, in green the post-peak training and in yellow the validation set.

2.2.4 Windowing and Resampling

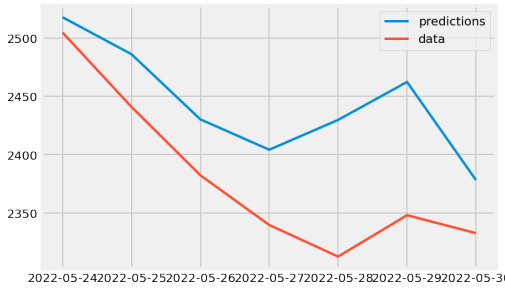
The dataset is then created by generating input windows of 14 days, in order to capture a bi-week trend, and choosing as output the prediction for new hospitalized and deceased after seven days. The last step before fitting the network consists in a random resampling of the training set. By choosing to include also older days in our training set we have solved the issue of numerosity, but introduced the one of unbalance. Specifically, we would like more recent training instances to be weighted more during training: as a solution we resample the tuples of our training dataset, with probabilities increased for recent observations.

3. Numerical results

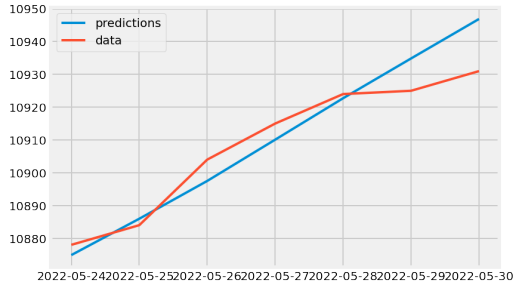
3.1. ARIMA predictions

To show the performance of our algorithm in the predicting phase, we report below the prediction results obtained with ARIMA model, referred to the days from 24-05-2022 to 30-05-2022. The metric used to assess the quality of these predictions is the mean absolute error:

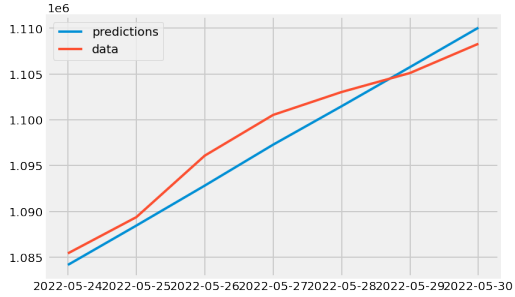
$$MAE = \frac{\sum_{t=1}^{7} |y(t) - \hat{y}(t)|}{7}$$



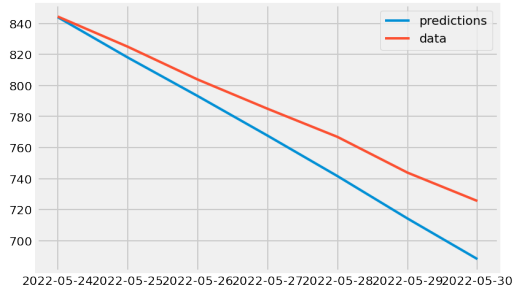
(a) New Infections Lazio. MAE = 64.06



(b) Deceased Sicilia. MAE = 6.22



(c) Recovered Sicilia. MAE = 1807.36

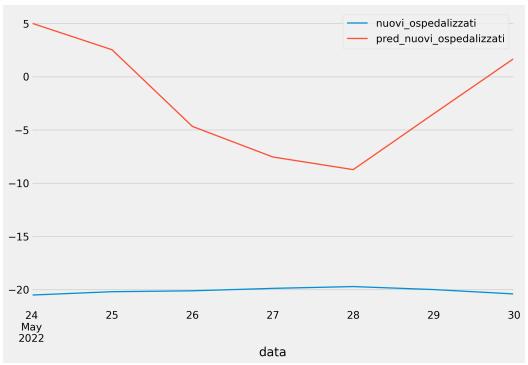


(d) Hospitalised Lombardia. MAE = 18.22

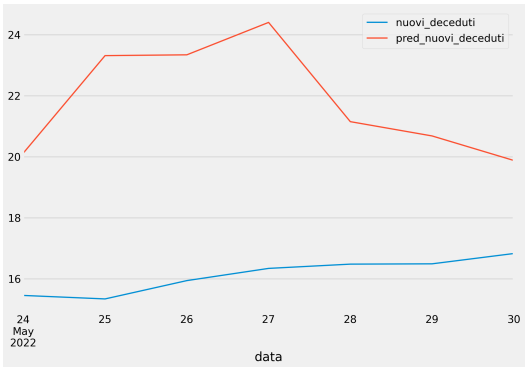
As we can see the algorithm is quite good in predicting the weekly trend for the new infections.
The MAE for the recovered in Sicily is larger than the ones of the other categories due to the order of magnitude of the cumulative data, thus these MAE should not be used to compare the goodness of the four predictions.

3.2. Neural Network Prediction

The performances of the Neural Networks are extremely poor when considering new positives and recovered, while their prediction are good for hospitalized and deceased. The reason is that Omicron changed significantly the evolution of the first two quantities, with a sharp increase in magnitude even after the peak. On the other hand, due to the minor gravity of the infection, the number of hospitalized and deceased remains comparable both before and after the peak.

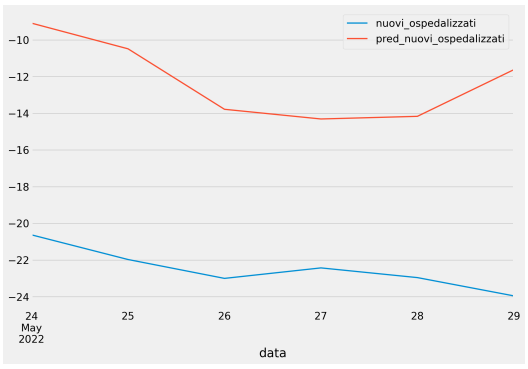


(a) Hospitalized Lombardia

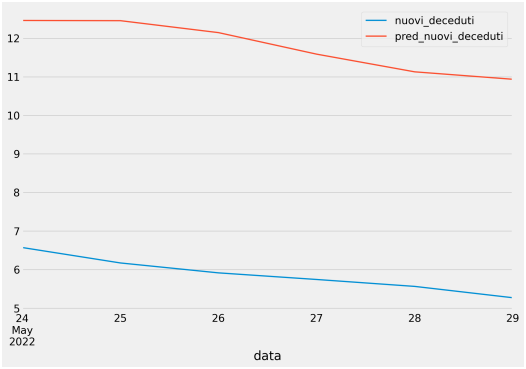


(b) Deceased Lombardia

Figure 8: Prediction for Lombardia, average MAE = 11.835

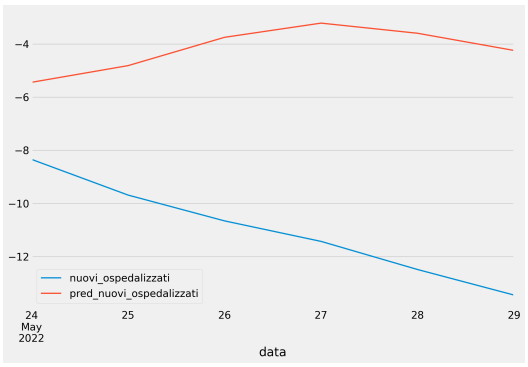


(a) Hospitalized Lazio

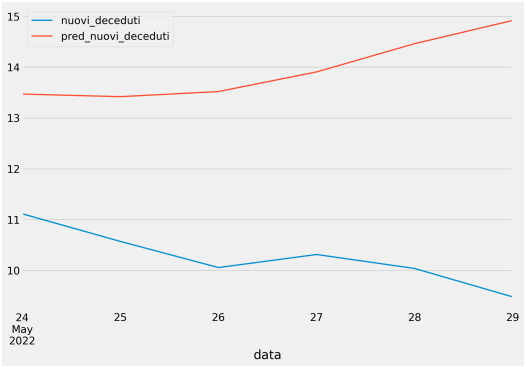


(b) Deceased Lazio

Figure 9: Prediction for Lazio, average MAE = 8.08



(a) Hospitalized Sicilia



(b) Deceased Sicilia

Figure 10: Prediction for Sicilia, average MAE = 5.26

3.3. Submission

For the submission we decided to select for each category the model that performs better, thus we used the ARIMA model to predict new infections and recovered, while the neural network to predict the hospitalised and deceased. As a result, for the day 18-05-2022 we obtained the following results:

18-05-2022 Predictions				
	hospitalised	deceased	new infections	recovered
Lombardia	1083	40219	4294	2671707
Lazio	942	11222	4095	1395747
Sicilia	714	10742	2567	1077160

Table 2: First Submission

while the true values of these categories were:

18-05-2022 True Values				
	hospitalised	deceased	new infections	recovered
Lombardia	949	40342	4325	2687117
Lazio	823	11268	2883	1395410
Sicilia	662	10824	2204	1060462

Table 3: First Submission - True Values

4. Conclusions

As we can see from the plots displayed in Section 3.1 and Section 3.2, the neural network performs better with hospitalised and deceased, while the ARIMA model with new infections and recovered. This is why the submission is done using both the models. It is worth to mention that, while the ARIMA model has good performance also for hospitalised and deceased, the neural network has poor perfomancces for new positives and recovered.

5. References

- The GitHub repository from which we extract the data can be found here